Applications of Information Theory in Image Processing

Image Denoising

Gadiel Seroussi

March 22, 2023

iDUDE: A DUDE-based Framework for Grayscale Image Denoising

The Discrete Universal DEnoiser (DUDE)

Summary

Given an input sequence $\mathbf{z} = z_1, z_2, \ldots, z_n$, run two passes on z^n.

Pass 1: For each i, determine the *context* $\mathcal{S}_i^{\mathbf{z}}$ of z_i in $\mathbf{z}$. Collect the empirical conditional distribution $\hat{P}_{\mathbf{z}}(z_i \,|\, \mathcal{S}_i^{\mathbf{z}})$.

Pass 2: For each i,

- *estimate* the PMF $\hat{P}_{\mathbf{x}}(\, x_i \,|\mathcal{S}_i^{\mathbf{z}}\,)$ based on

  "channel inversion"

  $$\hat{P}_{\mathbf{x}}(\,\cdot\,|\mathcal{S}_i^{\mathbf{z}}\,) = \mathbf{\Pi}^{-T} \cdot \hat{P}_{\mathbf{z}}(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$$ (as col vectors).

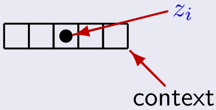
  Same $\hat{P}_{\mathbf{x}}(\,\cdot\,|\,\mathcal{S}_j^{\mathbf{z}})$ *for all* j *with* $\mathcal{S}_j^{\mathbf{z}} = \mathcal{S}_i^{\mathbf{z}}$.

- Given $\hat{P}_{\mathbf{x}}(\,\cdot\,|\mathcal{S}_i^{\mathbf{z}}\,)$, the observed $z_i = \alpha$, and the cost function Λ, compute a *cost-weighted MAP estimate* $\hat{x}_i$ of x_i:

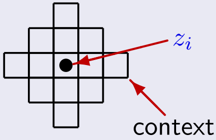
$$\hat{x}_i = \arg\min_{\hat{x} \in \mathcal{A}} \boldsymbol{\lambda}_{\hat{x}}^T \cdot \left(\left(\mathbf{\Pi}^{-T} \mathbf{m}[z^n, \mathbf{b}_i, \mathbf{c}_i] \right) \odot \boldsymbol{\pi}_\alpha \right)$$

Contexts $\mathcal{S}_i^{\mathbf{z}}$

1-dimensional (string)



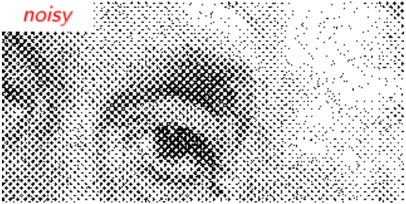
2-dimensional (image)

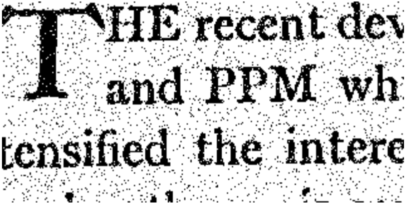


generally non-causal

The basic DUDE works in practice (sometimes)

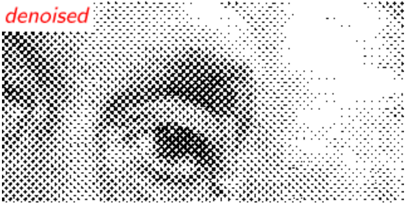
The basic scheme has been shown to work well for a variety of data types, including *text* and *binary images*.

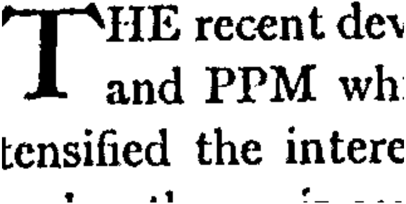




The basic DUDE works in practice (sometimes)

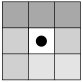
The basic scheme has been shown to work well for a variety of data types, including *text* and *binary images*.





The basic DUDE does not work in practice (sometimes)

Things get complicated when alphabets are large, e.g. grayscale images with $|\mathcal{A}| = 256$.



Even with a simple 3×3 template, the number of possible context patterns is $>1.8 \cdot 10^{19}$: *for images of practical size, contexts will seldom repeat, empirical distributions will be trivial, DUDE will do nothing.**

* *"almost hopeless"* [Buades, Coll, Morel: "A review of image denoising algorithms, with a new one," 2005].

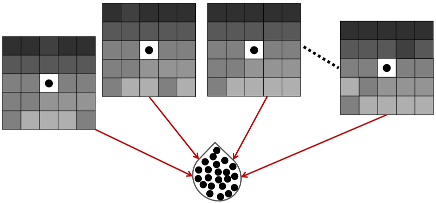
many names for one problem: statistics dilution, "curse of dimensionality," sparse contexts, *model cost*, *rate of convergence* (to best performance).

Lessons from *lossless image compression*:

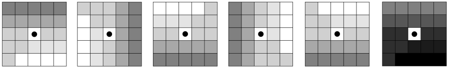
- Don't ask the algorithm to learn *what you already know*: use *prior knowledge*, be as universal as necessary, but not more
- *Aggregate* statistics from contexts that are "close" and produce "similar" conditional distributions: *context quantization*

Assumptions and tools for *clean* image modeling

- *Smoothness:* Contexts that are close as vectors (e.g., in L_2) tend to produce similar conditional distributions and can be merged



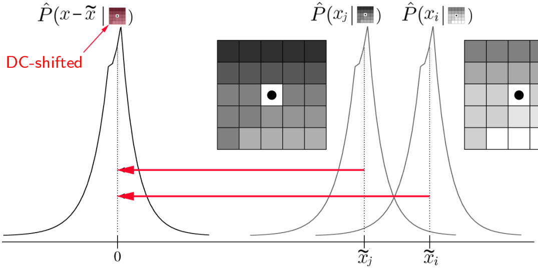
- *Symmetries:* Contexts that are similar up to spatial and black/white symmetries can be merged



$\Longrightarrow$ bring contexts to *canonical representations* modulo these symmetries

Assumptions and tools for *clean* image modeling

- *Prediction:* If $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{x}})$ is a good predictor of the center sample of $\mathcal{S}_i^{\mathbf{x}}$, then $P(\,\cdot\,|\mathcal{S}_i^{\mathbf{x}})$ concentrates around $\tilde{x}_i \implies$ model *prediction errors*: $P(x_i - \tilde{x}_i \,|\, \mathcal{S}_i^{\mathbf{x}})$
- *DC invariance:* Contexts that are similar up to a constant shift in brightness will have similar conditional distributions, up to a shift in the support (goes well with previous item)

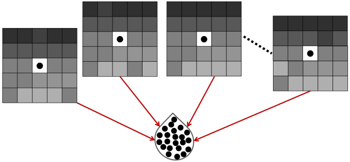


$\implies$ use *differential, DC-invariant, representations* for contexts (e.g., based on gradients)

Noisy complications

Assumptions break for noisy images, especially with *impulse* channels, e.g., the S&P channel.

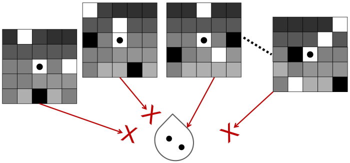
Smoothness



Noisy complications

Assumptions break for noisy images, especially with *impulse* channels, e.g., the S&P channel.

Smoothness



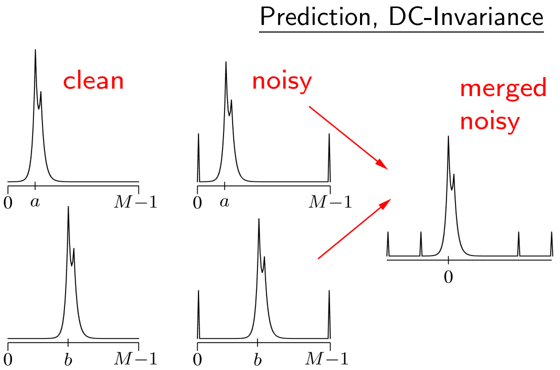
S&P channel:

$$a \to \begin{cases} 0 & (\delta/2) \\ a & (1-\delta) \\ M-1 & (\delta/2) \end{cases}$$

contexts that were close in the clean image are not so in the noisy one

Noisy complications

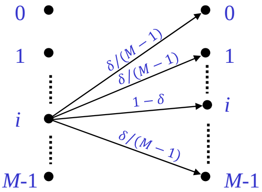
Assumptions break for noisy images, especially with *impulse* channels, e.g., the S&P channel.



Applying channel inversion to shifts of this distribution will not give good estimates of either clean distribution

Noisy complications (cont.)

- Impulse channels:
  - Noisy samples do not satisfy smoothness or DC-invariance
  - Noisy samples can throw predictor off
- Another type of impulse channel: *M-ary symmetric channel* (MSC)



$$P(b|a) = \begin{cases} 1-\delta & a = b, \\ \delta/(M-1) & a \neq b \end{cases}$$

- "Continuous" channels such as the Gaussian channel tend to preserve the assumptions better.

  $x \to \left[\, x + \mathcal{N}(0, \sigma^2) \,\right]$ (with saturation at $0,\ M{-}1$)

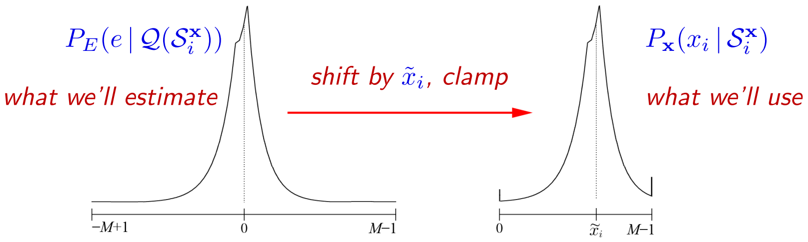
  in general, we need a second look at our image assumptions and model

A formal model for *clean* images

- $\tilde{x} : \mathcal{A}^{|\mathcal{S}|} \to \mathcal{A}$ predictor for x_i: $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{x}})$
- $\mathcal{Q} : \mathcal{A}^{|\mathcal{S}|} \to \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_K\}$ *context quantization* function: maps contexts $\mathcal{S}$ to *clusters* (or *conditioning classes*) $\mathcal{Q}(\mathcal{S})$
  - *DC-invariant:* $\mathcal{Q}(\mathcal{S}) = \mathcal{Q}(\mathcal{S}')$ whenever $\mathcal{S} = \mathcal{S}'$ up to DC shift
  - Number of clusters K controls model cost/performance trade-off
- A set of *cluster-conditioned* distributions $P_E(e \mid \mathcal{Q}_\kappa)$, $1 \le \kappa \le K$, of *prediction error values* $e \in \mathcal{A}_E \triangleq \{-M+1, -M+2, \ldots, M-1\}$

A formal model for *clean* images

- The model assumes that image samples are generated as follows:
  - e_i is drawn independently with probability $P_E(e_i \,|\, \mathcal{Q}(\mathcal{S}_i^{\mathbf{x}}))$
  - $x_i = \big[\, e_i + \tilde{x}(\mathcal{S}_i^{\mathbf{x}}) \,\big]_{0..M-1}$ ($[x]_{0..M-1}$ denotes *clamping* to $[0..M-1]$)



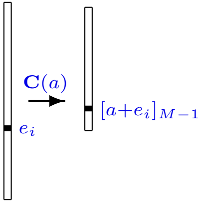
Sampling and estimation

- Represent $x \in \mathcal{A}$ by an indicator M-vector $\mathbf{u}_M^x = [\underbrace{0\,0\,\ldots\,0}_{x}\,1\,0\,\ldots\,0]^T$

- Represent $e \in \mathcal{A}_E$ by an indicator $(2M-1)$-vector $\mathbf{u}_{2M-1}^e = [\underbrace{0\,0\,\ldots\,0}_{e+M-1}\,1\,0\,\ldots\,0]^T$ (with some notation abuse)

- For $a \in \mathcal{A}$, define the $M\times(2M-1)$ *shift and clamp* matrix

$$\mathbf{C}(a) = \begin{array}{c}
\begin{array}{ccc} \overleftrightarrow{\quad M-1-a \quad} & \qquad\qquad & \overleftrightarrow{\quad a \quad} \end{array} \\
\left[\begin{array}{cccc|c|cccc}
1 & 1 & \cdots & 1 & & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & & \vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots & \mathbf{I}_{M\times M} & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & \textit{identity} & 1 & 1 & \cdots & 1
\end{array}\right] \\
\begin{array}{ccccccccc} -M+1 & -M+2 & \cdots & -a-1 & -a \;\cdots\; M-1-a & M-a & & \cdots & M-1 \end{array}
\end{array},$$



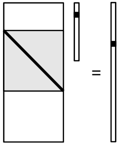
- $$x_i = \left[\, e_i + \tilde{x}_i \,\right]_{0..M-1} \quad \Longleftrightarrow \quad \mathbf{u}_M^{x_i} = \mathbf{C}(\tilde{x}_i)\cdot\mathbf{u}_{2M-1}^{e_i}$$

- Given $\tilde{x}_i(\mathcal{S}_i^{\mathbf{x}})$ and $\mathcal{Q}_\kappa = \mathcal{Q}(\mathcal{S}_i^{\mathbf{x}})$, observing x_i informs us about a window of length M in the support of $P_E(\cdot \,|\, \mathcal{Q}_\kappa)$

Estimation of $P_E(\cdot \mid \mathcal{Q}_\kappa)$

To estimate $P_E(\cdot \mid \mathcal{Q}_\kappa)$, we maintain a $(2M-1)$-vector statistic $\mathbf{e}_\kappa$, initialized to $\mathbf{0}$. Upon observing x_i, we perform the update

$$\mathbf{e}_\kappa \leftarrow \mathbf{e}_\kappa + \mathbf{M}(\tilde{x}_i) \cdot \mathbf{u}_M^{x_i}$$



where $\mathbf{M}(\tilde{x}_i)$ is a $(2M-1) \times M$ *estimation matrix*.
At this point, the obvious choice is

$$\mathbf{M}(\tilde{x}_i) = \left[\mathbf{0}_{M \times (M-1-\tilde{x}_i)} \, \middle| \mathbf{I}_M \middle| \, \mathbf{0}_{M \times \tilde{x}_i} \right]^T ,$$

equivalent to incrementing the location corresponding to $e_i = x_i - \tilde{x}_i$, but other choices are possible.

Theorem

$$\begin{aligned} \hat{\mathbf{P}}_E(\mathcal{Q}_\kappa) &= \left(\sum_{i:\mathcal{Q}_i^{\mathbf{x}} = \mathcal{Q}_\kappa} \mathbf{M}(\tilde{x}_i)\mathbf{C}(\tilde{x}_i) \right)^{-1} \cdot \left(\sum_{i:\mathcal{Q}_i^{\mathbf{x}} = \mathcal{Q}_\kappa} \mathbf{M}(\tilde{x}_i)\, \mathbf{u}_M^{x_i} \right) \\ &\triangleq \mathbf{R} \cdot \left(\sum_{i:\mathcal{Q}_i^{\mathbf{x}} = \mathcal{Q}_\kappa} \mathbf{M}(\tilde{x}_i)\, \mathbf{u}_M^{x_i} \right) \end{aligned}$$

is an unbiased estimator of $P_E(\cdot \mid \mathcal{Q}_\kappa)$ ($\mathbf{R}$ is a *normalization matrix*).

Model estimation for *clean* images

▸ noisy

1. *Initialization.* Initialize to zero a histogram, $\mathbf{e}_\kappa$, of prediction error residual occurrences for each context cluster $\mathcal{Q}_\kappa,\ 1 \le \kappa \le K$.

2. *Statistics collection.* For each index $i \in V_{m\times n}$:

   a. Set $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{x}})$, the predicted value for x_i.
   b. Set $\bar{\mathcal{S}}_i^{\mathbf{x}} = \mathcal{D}(\mathcal{S}_i^{\mathbf{x}})$, the differential representation of $\mathcal{S}_i^{\mathbf{x}}$.
   c. Set $\mathcal{C}_i^{\mathbf{x}} = \mathcal{C}(\bar{\mathcal{S}}_i^{\mathbf{x}})$, the canonical representation of $\bar{\mathcal{S}}_i^{\mathbf{x}}$.
   d. Set $\mathcal{Q}_i^{\mathbf{x}} = \mathcal{Q}(\mathcal{C}_i^{\mathbf{x}})$, the conditioning class of $\mathcal{S}_i^{\mathbf{x}}$.
   e. Set $\mathbf{e}_\kappa \leftarrow \mathbf{e}_\kappa + \mathbf{M}(\tilde{x}_i)\,\mathbf{u}_M^{x_i}$ for κ such that $\mathcal{Q}_\kappa = \mathcal{Q}_i^{\mathbf{x}}$.

3. *Normalization.* For each κ, normalize $\mathbf{e}_\kappa$ to obtain $\hat{\mathbf{P}}_E(\cdot|\mathcal{Q}_\kappa)$.

4. *Conditional distributions for individual contexts.* For each index $i \in V_{m\times n}$:

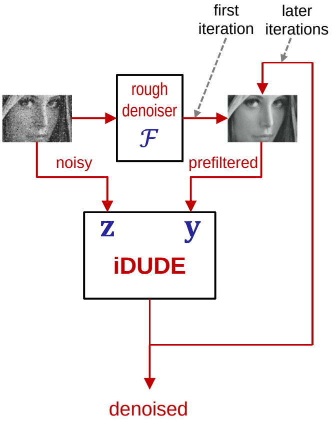
   a. Set $\tilde{x}_i$, $\mathcal{S}_i^{\mathbf{x}}$, and $\mathcal{Q}_i^{\mathbf{x}}$ as in Step 2 above.
   b. Set $\hat{\mathbf{P}}_X(\cdot|\mathcal{S}_i^{\mathbf{x}}) = \mathbf{C}(\tilde{x}_i)\,\hat{\mathbf{P}}_E(\cdot|\mathcal{Q}_i^{\mathbf{x}})$.

▸ definition of $\mathbf{C}$

How to deal with noise: avoid it (if you can!)

Try to carry out modeling operations in a cleaner domain, if possible

- Get contexts from a *prefiltered* version $\mathbf{y} = \mathcal{F}(\mathbf{z})$ of the noisy image
  - Contexts $\mathcal{S}_i^{\mathbf{y}}$, predictions $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{y}})$
  - Prefilter $\mathcal{F}$: a simple denoiser (e.g., median filter), or a previous application of the DUDE (*iterative DUDE*). *Counts for stats are still taken from the original noisy image* $\mathbf{z}$.
- Model prediction errors in *the clean image*. Estimate $\hat{P}_{\mathbf{x}}(\cdot\,|\,\cdot)$ *directly*, without going first through $\hat{P}_{\mathbf{z}}(\cdot\,|\,\cdot)$ and then inverting the channel.
  - *Assumption:* $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{y}})$ is a good predictor of x_i.



Channel inversion: a different view

Say the sequence observed in a given context $\mathbf{b} \bullet \mathbf{c}$ is

$$a^n = a_1, a_2, a_3, \ldots, a_n,$$

and let $\mathbf{m}$ be the histogram of occurrence of symbols in a^n, We have

$$\mathbf{m} = u_M^{a_1} + u_M^{a_2} + u_M^{a_3} + \cdots + u_M^{a_n}$$

and

$$\begin{aligned} \Pi^{-T}\mathbf{m} &= \Pi^{-T}u_M^{a_1} + \Pi^{-T}u_M^{a_2} + \Pi^{-T}u_M^{a_3} + \cdots + \Pi^{-T}u_M^{a_n} \\ &= \boldsymbol{\pi}_{a_1}^{-T} + \boldsymbol{\pi}_{a_2}^{-T} + \boldsymbol{\pi}_{a_3}^{-T} + \cdots + \boldsymbol{\pi}_{a_n}^{-T} \end{aligned}$$

where $\boldsymbol{\pi}_i^{-T}$ is the i-th column of Π^{-T} (transpose of the i-th row of Π^{-1}).

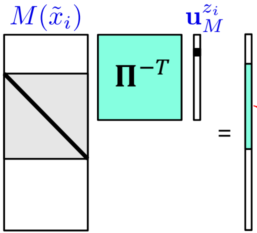
We can interpret this as computing the channel inversion *symbol by symbol*.

- Estimation of $P_E(\cdot \,|\, \mathcal{Q}_\kappa)$ (dist. of prediction errors $e_i = x_i - \tilde{x}_i$):
  Upon observing z_i, update the statistic $\mathbf{e}_\kappa$ with

$$\mathbf{e}_\kappa \to \mathbf{e}_\kappa + \underbrace{\mathbf{M}(\tilde{x}_i)\mathbf{\Pi}^{-T}}_{\mathbf{M}'(\tilde{x}_i)} \mathbf{u}_M^{z_i}$$

observations from the noisy image

estimation matrix M



shifted row of $\mathbf{\Pi}^{-1}$ (transposed)

- channel inversion done *on a sample-by-sample basis* (in practice, can still be done with scalar increments and some post-processing).
- $\hat{P}_{\mathbf{x}}(\cdot \,|\, \mathcal{S}_i^{\mathbf{y}})$ recovered from $\hat{\mathbf{P}}_E(\cdot|\mathcal{Q}_\kappa)$ by shifting and clamping, using the *observable* prediction $\tilde{x}_i$.

Model estimation for *noisy* images

The image $\mathbf{x}$ from the clean image procedure decouples into three images: $\mathbf{x}$ (unknown), $\mathbf{z}$ (noisy), $\mathbf{y}$ (prefiltered).

clean

1. *Initialization.* Initialize to zero a histogram, $\mathbf{e}_\kappa$, of prediction error residual occurrences for each context cluster $\mathcal{Q}_\kappa,\ 1 \le \kappa \le K$.

2. *Statistics collection.* For each index $i \in V_{m\times n}$:

   a. Set $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{y}})$, the predicted value for x_i.
   b. Set $\bar{\mathcal{S}}_i^{\mathbf{y}} = \mathcal{D}(\mathcal{S}_i^{\mathbf{y}})$, the differential representation of $\mathcal{S}_i^{\mathbf{y}}$.
   c. Set $\mathcal{C}_i^{\mathbf{y}} = \mathcal{C}(\bar{\mathcal{S}}_i^{\mathbf{y}})$, the canonical representation of $\bar{\mathcal{S}}_i^{\mathbf{y}}$.
   d. Set $\mathcal{Q}_i^{\mathbf{y}} = \mathcal{Q}(\mathcal{C}_i^{\mathbf{y}})$, the conditioning class of $\mathcal{S}_i^{\mathbf{y}}$.
   e. Set $\mathbf{e}_\kappa \leftarrow \mathbf{e}_\kappa + \mathbf{M}'(\tilde{x}_i)\,\mathbf{u}_M^{z_i}$ for κ such that $\mathcal{Q}_\kappa = \mathcal{Q}_i^{\mathbf{y}}$.

3. *Normalization.* For each κ, normalize $\mathbf{e}_\kappa$ to obtain $\hat{\mathbf{P}}_E(\cdot|\mathcal{Q}_\kappa)$.

4. *Conditional distributions for individual contexts.* For each index $i \in V_{m\times n}$:

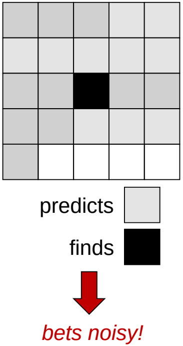
   a. Set $\tilde{x}_i$, $\mathcal{S}_i^{\mathbf{y}}$, and $\mathcal{Q}_i^{\mathbf{y}}$ as in Step 2 above.
   b. Set $\hat{\mathbf{P}}_X(\cdot|\mathcal{S}_i^{\mathbf{y}}) = \mathbf{C}(\tilde{x}_i)\,\hat{\mathbf{P}}_E(\cdot|\mathcal{Q}_i^{\mathbf{y}})$.

Iteration monitoring

- Prefiltering and iteration may violate some of the DUDE's basic independence assumptions.
  - *the reason:* the center sample z_i may have participated in the prefiltering of its neighbors.

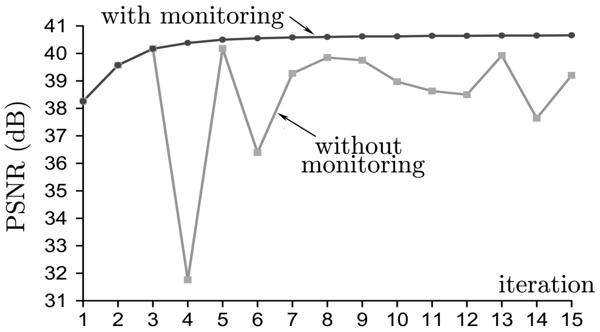
  This can cause performance deterioration and severe instability over iterations.

- *Solution approach:*
  - Estimate the fraction of *corrupted* sample values that occur in each cluster $\mathcal{Q}_\kappa$.
    - For each possible noisy value c, count the fraction of times c occurs in $\mathcal{Q}_\kappa$ when the corresponding prediction is *far* from c (e.g., black pixel in light background).
  - If the estimated noise fraction differs significantly from what's expected from the channel parameters, stop iterations for samples in $\mathcal{Q}_\kappa$ (i.e.: retain the last prefiltered value before the violation).



Iteration monitoring (cont.)

Example of iteration monitoring for a grayscale image affected by S&P noise (10%).



Additional details, features, tricks

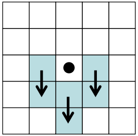
- *Bias cancelation*. As in lossless compression, we add an adaptive component to the predictor to neutralize context-dependent biases in the fixed predictor.
  - Contexts are first classified into N *prediction classes* (or clusters) $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_N$.
  - Bias statistics are collected in each prediction class $\mathcal{R}_r$, and a bias correction ϵ_r is computed for the class after the first DUDE pass. *Bias is measured relative to the prefiltered value y_i.*
  - Prediction classes are then *re-clustered* into the *conditioning classes* $\mathcal{Q}_\kappa$, $1 \le \kappa \le K$.
  - If $\mathcal{S}_i^{\mathbf{y}} \in \mathcal{R}_r$, then the final prediction value for x_i is

$$\hat{x}_i = \tilde{x}_i + \epsilon_r,$$

    which is used to compute the prediction error e_i used for the statistics of the corresponding $\mathcal{Q}_\kappa$.

Additional details, features, tricks (cont.)

- *Prediction and context classification.*
  - In the current implementation of iDUDE, prediction and context classification are based on the 5×5 window surrounding a sample.
  - Both use *wing gradients* (8 directions), from which some edge detection capability and a measure of context *activity* (or energy) are derived. In addition, a signature of the context *texture* is used in context classification.



S wing

- *Channel inversion.*
  - For the S&P and M-ary symmetric channel, the matrices $\mathbf{\Pi}$ are invertible and well conditioned. The inverse $\mathbf{\Pi}^{-1}$ is used as described so far.
  - For the Gaussian channel, the matrix $\mathbf{\Pi}$ is invertible but very badly conditioned—for practical purposed, we can consider it singular. Various approaches can be used to approximate channel inversion
    - Find a solution $\mathbf{P_x}$ that minimizes $||\hat{\mathbf{P}}_\mathbf{z} - \mathbf{\Pi}\hat{\mathbf{P}}_\mathbf{x}||$ under some numerical constraints.
    - Use *parametric* representations for $P_\mathbf{x}$ and $P_\mathbf{z}$ (e.g., Laplacian and Gaussian/Laplacian convolution, respectively), and estimate the parameters of $P_\mathbf{x}$.

iDUDE: a framework for grayscale image denoising

- *Why "framework"?* iDUDE is a general architecture for a denoising system incorporating the basic DUDE principles
  - estimation of context-conditioned *clean sample* probability distributions
  - an optimal Bayes *denoising rule* based on the estimated distributions and the given loss function

  together with a set of basic assumptions on grayscale images.
- The framework is instantiated for different noise types or image characteristics, by choosing specific embodiments for different algorithmic components
  - knowledge about the noisy channel, in the form of the *channel transition matrix*, drives the choice of an appropriate channel inversion method
  - it also dictates the choice of an appropriate *prefilter*
  - knowledge about the noise, the image type, and their interaction, drives the design of the context model, including
    - *context classification and aggregation* strategies
    - choice of conventional *image predictors*

Denoising examples and results

- iDUDE was instantiated for three types of noise
  - S&P
  - M-ary symmetric
  - Zero-mean additive white Gaussian (AWGN): $z_i = x_i + \eta_i$, where

$$\eta_i \sim f_\sigma(u) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-u^2/(2\sigma^2)\right)$$

probability density function

- Loss function: *Peak Signal to Noise Ratio* (PSNR):

MSE=Mean Squared Error, RMSE=Root MSE

$$\mathsf{MSE} = \frac{1}{m \cdot n} \sum_i (\hat{x}_i^{\mathsf{DUDE}} - x_i)^2, \quad \mathsf{RMSE} = \sqrt{\mathsf{MSE}}$$

$$\mathsf{PSNR} = 20 \log_{10} \frac{M-1}{\mathsf{RMSE}} \quad \text{(in dB)}$$

Bottom line: iDUDE significantly surpasses the published state of the art for *impulse* channels. It is competitive with very respectable schemes, but loses to the best, on the Gaussian channel—work in progress.

Gaussian channel

- Prefiltering and iteration have little effect (at most 2 iterations)
- Two types of context classification schemes
  - one based on gradients/texture as used for S&P and MSC (fast)
  - one based on vector quantization using LBG (Linde-Buzo-Gray), better for low SNR, but slower
- channel inversion done through an approximate ML estimation of a TSGD for the clean prediction errors
- needs improvement!

Denoising example: M-ary symmetric channel

Clean image

Denoising example: M-ary symmetric channel

Noisy: M-ary symmetric 30% RMSE = 48.2 PSNR = 8.6dB

Denoising example: M-ary symmetric channel

Denoised: M-ary symmetric 30% RMSE = 14.5 PSNR = 29.5dB

Denoising example: Salt and Pepper noise

Clean image

Denoising example: Salt and Pepper noise

Noisy: S&P 50% RMSE = 107.6 PSNR = 6.6dB

Denoising example: Salt and Pepper noise

Denoised: RMSE = 7.5 PSNR = 31.8dB

Denoising example: Salt and Pepper noise

Clean image

Image set

classical images 512×512



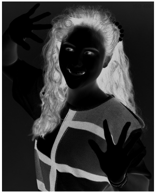
very small images (24) 384×256





larger images 1524×1200 to 2048×2560






Sample of denoiser parameters

S&P										M-ary symmetric						
	v. small			small			large				small			large		
δ	R	K	T	R	K	T	R	K	T	δ	R	K	T	R	K	T
10%	10	4	8	10	4	14	15	32	16	10%	15	4	14	15	8	16
30%	10	4	8	10	4	14	15	32	16	20%	15	4	14			
50%	10	4	8	10	4	14	20	32	16	30%	15	4	10	15	8	16
70%	20	4	8	20	4	14	20	32	14	40%	20	4	9			
										50%	20	4	8	20	16	8

$K \cdot 2^T =$ number of prediction classes, $K =$ number of conditioning classes, $R =$ number of iterations

Gaussian							
		LBG				WGT	
		small		large		all	
σ	R	K	N	K	N	K	T
5	1	32	256	96	256	32	6
20	2	32	192	32	192		

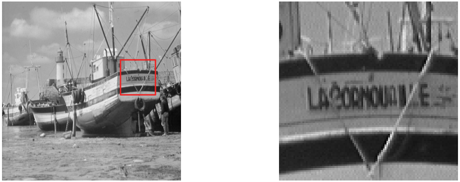
$K \cdot 2^T$ or $K \cdot N =$ number of prediction classes,
$K =$ number of conditioning classes,
$R =$ number of iterations

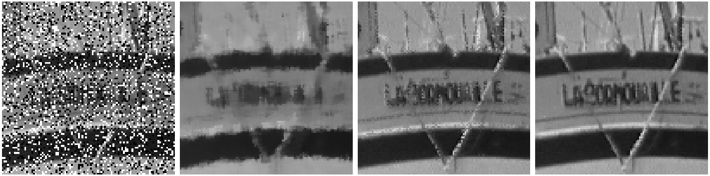
Results for the S&P channel: Comparisons

image	$\delta = 10\%$			$\delta = 30\%$			$\delta = 50\%$			$\delta = 70\%$		
	MSM	IMSM	iDUDE	MSM	IMSM	iDUDE	MSM	IMSM	iDUDE	MSM	IMSM	iDUDE
Lena	40.1	40.4	45.2	34.1	35.2	39.7	27.4	32.0	36.3	16.7	29.1	32.8
Boat	36.3	36.5	41.0	30.6	31.2	35.3	25.5	28.3	32.0	16.4	25.7	28.9
Barbara	32.6	33.0	38.7	27.4	28.3	31.7	23.4	26.0	27.7	15.8	24.2	24.7
Tools	25.6	25.2	31.8	22.1	22.2	26.9	19.2	20.1	23.5	14.1	18.5	20.6
Toolsk	27.1	26.8	31.0	23.6	23.8	26.4	20.0	21.7	23.6	12.9	20.2	21.2
Womank	34.0	33.9	40.7	30.0	30.3	34.9	24.6	27.9	31.2	14.3	26.1	28.1
Bike	31.2	31.3	39.4	26.5	27.4	33.1	22.4	24.7	29.0	15.0	22.1	25.1

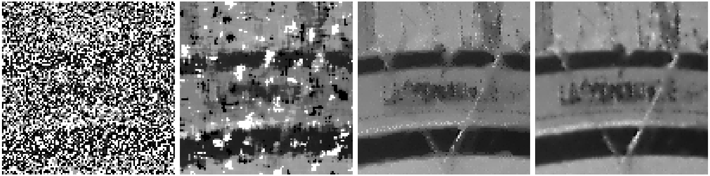
image/ δ	MSM	IMSM	CHN05	iDUDE
Set_{24}				
10%	36.3	36.5	40.4	40.9
30%	30.6	31.4	34.5	35.1
50%	25.0	28.4	31.1	31.6
70%	15.8	25.9	28.1	28.6
Lena*				
10%	38.9	39.2	42.3	44.8
30%	32.9	33.9	35.6	38.8
50%	26.4	30.8	32.3	35.4
70%	16.1	28.0	29.3	31.7

Results for the S&P channel: Visuals





(a) Noisy, $\delta = 30\%$ (b) MSM (30.6dB) (c) IMSM (31.2dB) (d) iDUDE (35.3dB)



(e) Noisy, $\delta = 70\%$ (f) MSM (16.4dB) (g) IMSM (25.7dB) (h) iDUDE (28.9dB)

Results for the MSC: Comparisons and Visuals

image	$\delta = 10\%$		$\delta = 30\%$		$\delta = 50\%$	
	MED	iDUDE	MED	iDUDE	MED	iDUDE
Boat	26.9	33.9	25.8	29.6	23.5	26.6
Barbara	23.1	29.9	22.7	25.4	21.2	23.5
Tools	18.9	26.9	18.4	22.3	17.1	19.2
Bike	23.4	31.1	22.4	26.0	19.9	22.2

image: Lena			
δ	MED	ROAD	iDUDE
10%	30.0	–	39.8
20%	30.1	35.0	36.9
30%	29.3	33.2	34.4
40%	27.8	31.4	32.8
50%	25.5	29.4	30.4

(a) Noisy, $\delta = 20\%$ (16.2dB)

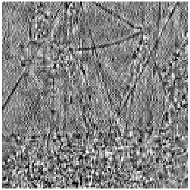
(b) MED (30.1dB)

(c) iDUDE (36.9dB)

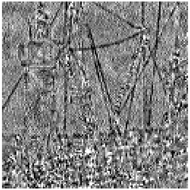
Results for the Gaussian channel: comparisons and visuals

image	$\sigma = 5$				$\sigma = 20$		
	BM3D	NLM	iDUDE	iDUDE$^{\mathsf{F}}$	BM3D	NLM	iDUDE
Lena	38.7	37.7	38.0	37.8	33.0	31.3	31.3
Boat	37.2	36.1	36.6	36.3	30.9	29.6	29.4
Barbara	38.3	37.1	36.9	36.2	31.7	30.1	28.6
Tools	36.3	35.5	35.9	35.7	28.5	27.2	27.0
Bike	38.8	37.6	37.7	37.4	32.1	30.8	29.8

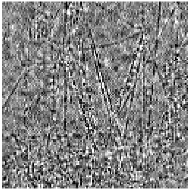
(a) Clean



(b) BM3D (33.8 dB)



(c) NLM (32.9 dB)



(d) iDUDE (32.9 dB)

error images ($\sigma = 10$; zero error = gray level 128)